UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/554,323	07/13/2006	Ian George Griffiths	GB920030030US1	4569

35525 7590 02/24/2009
IBM CORP (YA)
C/O YEE & ASSOCIATES PC
P.O. BOX 802333
DALLAS, TX 75380

EXAMINER	
BROPHY, MATTHEW J	
ART UNIT	PAPER NUMBER
2191	
NOTIFICATION DATE	DELIVERY MODE
02/24/2009	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

ptonotifs@yeeiplaw.com

Office Action Summary	**Application No.** 10/554,323	**Applicant(s)** GRIFFITHS ET AL.	
	Examiner MATTHEW J. BROPHY	**Art Unit** 2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *24 November 2008*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-5,14,20 and 21* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-5,14,20 and 21* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1. This office action is in response to the amendment filed November 24, 2008.

2. Claims 1-5, 14, 20 and 21 are pending.

Response to Amendment

Claim Rejections - 35 USC § 102

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

3. Claims 1,2, 5,14, 20 and 21 are rejected under 35 U.S.C. 102(b) as being anticipated by USPN 5,966,702 Fresko et al hereinafter Fresko.

Regarding Claims 1, Fresko teaches: A data processing method for creating an executable file by combining a plurality of run units, the method comprising:

<u>identifying a first data entity and a second data entity, wherein both the first data entity and the second data entity are identified using an Assembler instruction, and wherein the Assembler instruction identifies character strings which are required to appear only once in the executable file;</u> **(Column 9, Lines 17-21, "In step 402, the pre-processor examines the constant pool tables of each class to determine the set of class file constants (such as strings and numerics, as well as others specific to the class file format) that can be shared between classes in "S."" i.e. determines which entities are "constants" see also Col. 8, Ln 10-14"Constant pool table 305 is a table of variable-length data structures <u>representing various string constants</u>, numerical constants, class names, field names, and other constants that are referred to within the ClassFile structure." See further Col. 9, Ln 3-5"Also, by**

implementing a shared constant table, entries in the constant table <u>need be fully resolved at most once.</u>”))

responsive to a determination that a first run unit to be added to the executable file comprises a first data entity set to a first value indicating that the first data entity is required to appear only once in the executable file **(Column 9, Lines 17-21, “In step 402, the pre-processor examines the constant pool tables of each class to determine the set of class file constants (such as strings and numerics, as well as others specific to the class file format) that can be shared between classes in "S."” i.e. determines which entities are “constants”),** determining whether the first data entity matches a second data entity set to a second value and included in a second run unit, wherein the second run unit comprises a run unit that was previously added to the executable file **(Column 9, Lines 21-23, “A shared constant pool table is created in step 403, with all duplicate constants determined from step 402.”)** *[here the examination of classes inherently is carried out sequentially, and thus the first class examined/inserted in the class file reads on “a run unit that was added previously”]*;

responsive to a determination that the first data entity matches the second data entity, adding the first run unit to the executable file [[but]] without the first data entity **(Column 9, Lines 23-35, “In step 404, the pre-processor removes the duplicate, shared constants from the individual constant pool tables of each class.”)**. ; and responsive to a determination that the first data entity does not match the second data entity, adding the first run unit to the executable file with the first data entity **(Column 9,**

Lines 23-35, "In step 404, the pre-processor removes the duplicate, shared constants from the individual constant pool tables of each class." i.e. if they are NOT duplicates, they are NOT removed from the constant pool of the class when the class file is added).

Regarding Claims 2, Fresko teaches: A method of claim 1 wherein the first data entity [[with]] matches the second data entity if the first value and second value are identical. **(Column 9, Lines 21-23, "A shared constant pool table is created in step 403, with all duplicate constants determined from step 402.").**

Regarding Claims 5, Fresko teaches: A method of claim 1 wherein the <u>determination that the first run unit to be added to the executable file comprises a first data entity set to a first value indicating that the first data entity is required to appear only once in the executable file</u> **(Column 9, Lines 17-21, "In step 402, the pre-processor examines the constant pool tables of each class to determine the set of class file constants (such as strings and numerics, as well as others specific to the class file format) that can be shared between classes in "S."" i.e. determines which entities are "constants"),** the step of locating a first data entity comprises locating a plurality of data entities in the first run unit **(Column 9, Lines 21-23, "A shared constant pool table is created in step 403, with all duplicate constants determined from step 402.")**; and creating the first data entity from the plurality of data entities **(Column 9, Lines 21-23, "A shared constant pool table is created in step 403, with all duplicate constants determined from step 402.").**

Regarding Claim 14, Fresko teaches: A method of claim 5 wherein the locating a plurality of data entities comprises locating a plurality of data entities using a key value by which each of the plurality of data entities is marked. **(Col. 9 Ln 55-57, "In one embodiment of the invention, a new constant type is defined with a corresponding constant type tag. The new constant type provides as its info[ ] element an index into the shared constant table.")**

Regarding Claim 21, Fresko teaches: A method of claim 1 wherein the Assembler instruction is a DL Assembler instruction. **(Column 9, Lines 17-21, "In step 402, the pre-processor examines the constant pool tables of each class to determine the set of class file constants (such as strings and numerics, as well as others specific to the class file format) that can be shared between classes in "S."" i.e. determines which entities are "constants" see also Col. 8, Ln 10-14"Constant pool table 305 is a table of variable-length data structures <u>representing various string constants</u>, numerical constants, class names, field names, and other constants that are referred to within the ClassFile structure." See further Col. 9, Ln 3-5"Also, by implementing a shared constant table, entries in the constant table <u>need be fully resolved at most once.</u>")** *[Fresko anticipates a DL Assembler where applicant's specification defines a DL Assembler as ""DL" is a new type of assembler instruction which is used in the preferred embodiment to denote a non-executable data entity which need only be included once in any file."]*

Regarding Claim 22, Fresko teaches: A method of claim 21, wherein the DL Assembler instruction is a type of Assembler instruction that denotes a non-executable data entity which needs only be included once in the executable file. **(Col. 3, Ln 18-35,**

"In FIG. 1, server 100 comprises Java development environment 104 for use in creating the Java class files for a given application. …Java source files 105 contain the programmer readable class definitions, including data structures, method implementations and references to other classes. Java source files 105 are provided to Java compiler 106, which compiles Java source files 105 into compiled ".class" files 107 that contain bytecodes executable by a Java virtual machine. Bytecode class files 107 are stored (e.g., in temporary or permanent storage) on server 100, and are available for download over network 101.")

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 3 and 4 are rejected under 35 U.S.C. 103(a) as being unpatentable over USPN 5,966,702 Fresko et al hereinafter Fresko in view of US PG Pub 2005/0114840 Ziedman hereinafter Ziedman.

Regarding Claim 3, Fresko teaches the limitations of Claim 1. Fresko does not teach: A method of claim 1 wherein the first data entity [[with]] matches the second data entity if the second value partially matches contains the first value. However, this limitation is taught by Ziedman **(e.g. "Partial Word Matching [0077] The "partial word**

matching" algorithm examines each identifier (non-keyword) word in the source code of one file of a file pair and finds all words that match a sequence within one or more non-keyword words in the other file of a file pair. Like the word matching algorithm, this one is also case insensitive. This algorithm is illustrated in FIG. 7. In part (a) 701, the non-keyword words from the two files are displayed. In part (b) 702, every word from one file that can be found as a sequence within a word from the other file is listed. So the identifier "abc" in file 1 can be found within identifiers "aabc", "abc1111111", and "abcxxxyz" in file 2. Note that identifier "pdq" is not listed in the array of partially matching words because it matches completely and was already considered in the word matching algorithm. Also note that identifier "x" is not listed in the array because 1-character words are ignored.") In addition it would have been obvious to one of ordinary skill in the art to combine the teachings of Fresko with the partial matching of Ziedman as the use of partial matching in Fresko's invention would further increase the memory space saved by Fresko's invention as Ziedman teaches "find[ing] all words that match a sequence within one or more non-keyword words in the other file of a file pair" (paragraph 77) such as in the string constants of Fresko.

Regarding Claim 4, Fresko further teaches: A method of claim 3 further comprising

<u>identifying a third data entity using an Assembler instruction that identifies character strings which are required to appear only once in the executable file;</u> **(Column 9, Lines 17-21, "In step 402, the pre-processor examines the constant pool tables**

of each class to determine the set of class file constants (such as strings and numerics, as well as others specific to the class file format) that can be shared between classes in "S."" i.e. determines which entities are "constants" see also Col. 8, Ln 10-14"Constant pool table 305 is a table of variable-length data structures <u>representing various string constants</u>, numerical constants, class names, field names, and other constants that are referred to within the ClassFile structure.")

determining whether matching the first data entity matches a third data entity included in a third run unit to be added to the executable file, wherein the third data entity is set to a third value indicating that the third data entity is required to appear only once in the executable file, **(Column 9, Lines 17-21, "In step 402, the pre-processor examines the constant pool tables of each class to determine the set of class file constants (such as strings and numerics, as well as others specific to the class file format) that can be shared between classes in "S."" i.e. determines which entities are "constants"),** and wherein the first data entity matches the third data entity a match is found if the third value contains the first value **(Column 9, Lines 21-23, "A shared constant pool table is created in step 403, with all duplicate constants determined from step 402.")**; responsive to a determination that the first data entity matches the third data entity, removing the first data entity from the executable file **(Column 9, Lines 23-35, "In step 404, the pre-processor removes the duplicate, shared constants from the individual constant pool tables of each class.")**; and adding the third data entity to the executable file **(Column 9, Lines 23-35, "In step 404,**

the pre-processor removes the duplicate, shared constants from the individual constant pool tables of each class.”).

.Response to Arguments

6. Applicant's arguments filed November 24, 2008 have been fully considered but they are not persuasive.

In remarks, Applicant argues:

In this case each and every feature of the presently claimed invention is not identically shown in Fresko, arranged as they are in the claims, and, accordingly, Fresko does not anticipate the claims. Specifically, Fresko does not teach or suggest "identifying a first data entity and a second data entity, wherein both the first data entity and the second data entity are identified using an Assembler instruction, and wherein the Assembler instruction identifies character strings which are required to appear only once in the executable file," as recited in amended independent claim 1. Page 5 of 8 Griffiths et al. - 10/554,323

Fresko is directed to a method and apparatus for pre-processing and packaging class files. Embodiments remove duplicate information elements from a set of class files to reduce the size of individual class files and to prevent redundant resolution of the information elements. Memory allocation requirements are determined in advance for the set of classes as a whole to reduce the complexity of memory allocation when the set of classes are loaded. The class files are stored in a single package for efficient storage, transfer and processing as a unit. In an embodiment, a pre-processor examines each class file in a set of class files to locate duplicate information in the form of redundant constants contained in a constant pool. The duplicate constant is placed in a separate shared table, and all occurrences of the constant are removed from the respective constant pools of the individual class files. During pre- processing, memory allocation requirements are determined for each class file, and used to determine a total allocation requirement for the set of class files. The shared table, the memory allocation requirements and the reduced class files are packaged as a unit in a multi-class file. Fresko does not teach or suggest "identifying a first data entity and a second data entity, wherein both the first data entity and the second data entity are identified using an Assembler instruction, and wherein the Assembler instruction identifies character strings which are required to appear only once in the executable file," as recited in amended independent claim 1.

Examiner's Response:

Examiner respectfully disagrees. In step 402, Fresko teaches identifying certain data entities (including strings) need to exist only once (i.e. in the shared table) when examining the source code. (See Col. 9, Ln. 17-21). While Fresko does not call step 402 a "DL Assembler instruction", it anticipates this element as described in new claim 22 because it identifies string constants in the source code (i.e. non-executable data entities) that must only appear once in the executable (i.e. compiled shared constant table). Therefore, this rejection is maintained.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to MATTHEW J. BROPHY whose telephone number is 571-270-1642. The examiner can normally be reached on Monday-Thursday 8:00AM-5:00 PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached on (571) 272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

MJB

2/4/2009

/Wei Y Zhen/
Supervisory Patent Examiner, Art Unit 2191